Introduction

Big Data Analytics

Presented by: Dr Sherin El Gokhy

Module 4 – Advanced Analytics - Theory and Methods

Module 4: Advanced Analytics – Theory and Methods

Part 2: Association Rules

During this part the following topics are covered:

- Association Rules mining
- Apriori Algorithm
- Prominent use cases of Association Rules
- Support and Confidence parameters
- Lift and Leverage
- Diagnostics to evaluate the effectiveness of rules generated
- Reasons to Choose (+) and Cautions (-) of the Apriori algorithm

Association Rules

Association Rules is another unsupervised learning method.

It is used to discover relationships within the data.

Some of the questions that can be answered using association rules:

- ✓ Which of my products tend to be purchased together?
- ✓ What do other people like this person tend to like/buy/watch?
- Discover "interesting" relationships among variables in a large database
  - "interesting" relationships are described in rules of the form “If X is observed, then Y is also observed"
  - The definition of "interesting“ varies with the algorithm used for discovery
- Not a predictive method; It is used to find similarities, relationships within your data.

Association Rules - Apriori

- Specifically designed for mining over transactions (**any change**) in databases
- Used over itemsets: sets of discrete variables that are linked together like:
  - Retail items that are purchased together
  - A set of tasks done in one day
  - A set of links clicked on by one user in a single session
- **The most commonly used algorithms for association rules: Apriori**

Apriori Algorithm - What is it?
Support

- Earliest of the association rule algorithms
- Apriori works on frequent itemset: a set of items (a set of links or a set of tasks) L that appears together "often enough“:
  - Formally often means: meets a minimum support criterion
  - **Support**: the percentage of transactions that contain L
- **Apriori Property: Any subset of a frequent itemset is also frequent**
- For example: If we define L as an itemset {shoes, purses} and we define our “support” as 50%. **If 50% of the transactions have this itemset , then we say L is a “frequent itemset”.**
- It is apparent that if 50% of itemsets have {shoes,purses} in them, then at least 50% of the transactions will have either {shoes} or {purses} in them. This is an **Apriori property.**
  - The subset has at least the support of its superset

Apriori Algorithm (Continued)
Confidence

- Iteratively grow the frequent itemsets from size 1 to size K..threshold (or until we run out of support).
- We start with all the frequent itemsets of size 1 (for example shoes, purses, hats etc.) first and determine the support. Then we start pairing them. We find the support for say {shoes, purses} or {shoes, hats} or {purses, hats}.
- Suppose we set our threshold as 50% we find those itemsets that appear in 50% of all transactions.
- We scan all the itemsets and "prune away" the itemsets that have less than 50% support (appear in less than 50% of the transactions), and keep the ones that have sufficient support.

Apriori Algorithm (Continued)
Confidence

- Apriori property provides the basis to prune over the transactions (search space) and to stop searching further if the support threshold criterion is **not** met.
- If the support criterion is met we grow the itemset and repeat the process until we have the specified number of items in an itemset or we run out of support.
- Frequent itemsets are used to find rules X->Y with a minimum confidence **(**rules such as X implies Y**)**.
- Confidence: The percentage of transactions that contain X, which also contain Y.
- For example if we have frequent itemset {shoes, purses, hats} and consider subsets {shoes, purses}. If 80% of the transactions that have {shoes, purses} also have {hats} we define Confidence for the rule that {shoes, purses} implies {hats} as 80%.

- Output of the apriori algorithm: The set of all rules X -> Y with minimum support and confidence.

Lift and Leverage

Lift measures how many times more often X and Y occur together than expected if they were statistically independent. It is a measure of how X and Y are really related rather than coincidentally happening together.

$$\text{Lift}(X \rightarrow Y) = \frac{\text{Support}(X \wedge Y)}{\text{Support}(X) * \text{Support}(Y)}$$

Leverage measures the difference in the probability of X and Y appearing together in the data set compared to what would be expected if X and Y were statistically independent (happening together by chance).

$$\text{Leverage}(X \rightarrow Y) = \text{Support}(X \wedge Y) - \text{Support}(X) * \text{Support}(Y)$$

Association Rules Implementations

- Market Basket Analysis
  - People who buy milk also buy cookies 60% of the time.
- Recommender Systems
  - "People who bought what you bought also purchased….".
- Discovering web usage patterns
  - People who land on page X click on link Y 76% of the time.

Use Case Example: Credit Records

Credit ID	Attributes
1	credit_good, female_married, job_skilled, home_owner, …
2	credit_bad, male_single, job_unskilled, renter, …

Given that minimum support: 50%

Frequent Itemset	Support
credit_good	70%
male_single	55%
job_skilled	63%
home_owner	71%
home_owner, credit_good	53%

The itemset {home_owner, credit_good} has minimum support.

The possible rules are

credit_good -> home_owner

and

home_owner -> credit_good

Computing Confidence and Lift

Suppose we have 1000 credit records:

	free_housing	home_owner	renter	total
credit_bad	44	186	70	**300**
credit_good	64	527	109	**700**
	108	**713**	**179**	

713 home_owners, 527 have good credit.

home_owner -> credit_good has confidence 527/713 = 74%

700 with good credit, 527 of them are home_owners

credit_good -> home_owner has confidence 527/700 = 75%

The lift of these two rules is

0.527 / (0.700*0.713) = 1.055

The lift being close to the value of 1 indicates that the rule is purely coincidental and with larger values of Lift (say >1.5) we may say the rule is “true” and not coincidental.

A Sketch of the Algorithm

- If L_k is the set of frequent k-itemsets:
  - Generate the candidate set C_{k+1} by joining L_k to itself
  - Prune out the (k+1)-itemsets that don't have minimum support Now we have L_{k+1}
- We know this catches all the frequent (k+1)-itemsets by the apriori property
  - a (k+1)-itemset can't be frequent if any of its subsets aren't frequent
- Continue until we reach k_{max} where max is the threshold, or run out of support
- From the union of all the L_k, find all the rules with minimum confidence

Step 1: 1-itemsets (L1)

- let min_support = 0.5
- 1000 credit records
- Scan the database
- Prune

Frequent Itemset	Count
credit_good	700
~~credit_bad~~	~~300~~
male_single	550
~~male_mar_or_wid~~	~~92~~
~~female~~	~~310~~
job_skilled	631
~~job_unskilled~~	~~200~~
home_owner	710
~~renter~~	~~179~~

Step 2: 2-itemsets (L2)

- Join L1 to itself
- Scan the database to get the counts
- Prune

Frequent Itemset	Count
~~credit_good, male_single~~	~~402~~
credit_good, job_skilled	544
credit_good, home_owner	527
~~male_single, job_skilled~~	~~340~~
~~male_single, home_owner~~	~~408~~
~~job_skilled, home_owner~~	~~452~~

Step 3: 3-itemsets

Frequent Itemset	Count
~~credit_good, job_skilled, home_owner~~	~~428~~

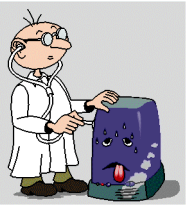
- We have run out of support.
- Candidate rules come from L2:
  - credit_good -> job_skilled
  - job_skilled -> credit_good
  - credit_good -> home_owner
  - home_owner -> credit_good

Finally: Find Confidence Rules

Rule	Set	Cnt	Set	Cnt	Confidence
IF credit_good THEN job_skilled	credit_good	700	credit_good AND job_skilled	544	544/700=77%
IF credit_good THEN home_owner	credit_good	700	credit_good AND home_owner	527	527/700=75%
IF job_skilled THEN credit_good	job_skilled	631	job_skilled AND credit_good	544	544/631=86%
IF home_owner THEN credit_good	home_owner	710	home_owner AND credit_good	527	527/710=74%

If we want confidence > 80%:
IF job_skilled THEN credit_good

Diagnostics



- Do the rules make sense?
  - What does the domain expert say?
- Make a "test set" from hold-out data:
  - Enter some market baskets with a few items missing (selected at random). Can the rules determine the missing items?
  - Remember, some of the test data may not cause a rule to fire.
- Evaluate the rules by lift or leverage.
  - Some associations may be coincidental (or obvious).

Apriori - Reasons to Choose (+) and Cautions (-)

Reasons to Choose (+)	Cautions (-)
Easy to implement	Requires many database scans
Uses a clever observation to prune the search space •Apriori property	Exponential time complexity
Easy to parallelize	Can mistakenly find spurious (or coincidental) relationships •Addressed with Lift and Leverage measures

Check Your Knowledge

1. What is the Apriori property and how is it used in the Apriori algorithm?
2. List three popular use cases of the Association Rules mining algorithms.
3. What is the difference between Lift and Leverage. How is Lift used in evaluating the quality of rules discovered?
4. Define Support and Confidence
5. How do you use a "hold-out" dataset to evaluate the effectiveness of the rules generated?

Consider the set of items is I = {milk, bread, butter, beer} and a small database of transactions containing the items (where 1 codes presence and 0 codes absence of an item in a transaction) is shown in the table below.

a) Apply Apriori algorithm (let the minimum support= 40%) to find all the frequent item se in the database.

b) Use these frequent item sets and the minimum confidence constraint (let the minimum confidence= 70%) to form the association rules.

Transaction ID	milk	Bread	butter	beer
1	1	1	0	0
2	0	1	1	0
3	0	0	0	1
4	1	1	1	0
5	0	1	0	0
6	1	0	0	0
7	0	1	1	1
8	1	1	1	1
9	0	1	0	1
10	1	1	0	0
11	1	0	0	0
12	0	0	0	0
13	1	1	1	0
14	1	0	1	0
15	1	1	1	1

Module 4: Advanced Analytics – Theory and Methods

Part 2: Association Rules - Summary

During this part the following topics were covered:

- Association Rules mining
- Apriori Algorithm
- Prominent use cases of Association Rules
- Support and Confidence parameters
- Lift and Leverage
- Diagnostics to evaluate the effectiveness of rules generated
- Reasons to Choose (+) and Cautions (-) of the Apriori algorithm

Lab Exercise 5 - Association Rules

- This Lab is designed to investigate and practice Association Rules.

  After completing the tasks in this lab you should be able to:

  - Use R functions for Association Rule based models

Lab Exercise 5 - Association Rules - Workflow

1. Set the Working Directory and install the “arules” and "arulesViz" package
2. Read in the Data for Modeling
3. Review Transaction data
4. Plot Transactions
5. Mine the Association Rules
6. Read in Groceries dataset
7. Mine the Rules for the Groceries Data
8. Extract Rules with Confidence > 0.8

Thanks